PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 :
G06F 17/30

A1

(11) International Publication Number: WO 98/25217

(43) International Publication Date: 11 June 1998 (11.06.98)

(21) International Application Number: PCT/US97/22943

(22) International Filing Date: 4 December 1997 (04.12.97)

(30) Priority Data:
08/760,691 4 December 1996 (04.12.96) US

(71) Applicant: QUARTERDECK CORPORATION [US/US]; Suite 234, 13160 Mindanao Way, Marina del Rey, CA 90292-9705 (US).

(72) Inventors: ULICNY, Brian, E.; 1221 1/2 Ozeta Terrace, Los Angeles, CA 90069 (US). JENSEN, John, B.; 1506 Palm Drive, Hermosa Beach, CA 90254 (US). ALLEN, Bradley, P.; 829 Loma Drive, Hermosa Beach, CA 90254 (US).

(74) Agent: YANG, Joseph; Skadden, Arps, Slate, Meagher & Flom, LLP, Suite 220, 525 University Avenue, Palo Alto, CA 94301 (US).

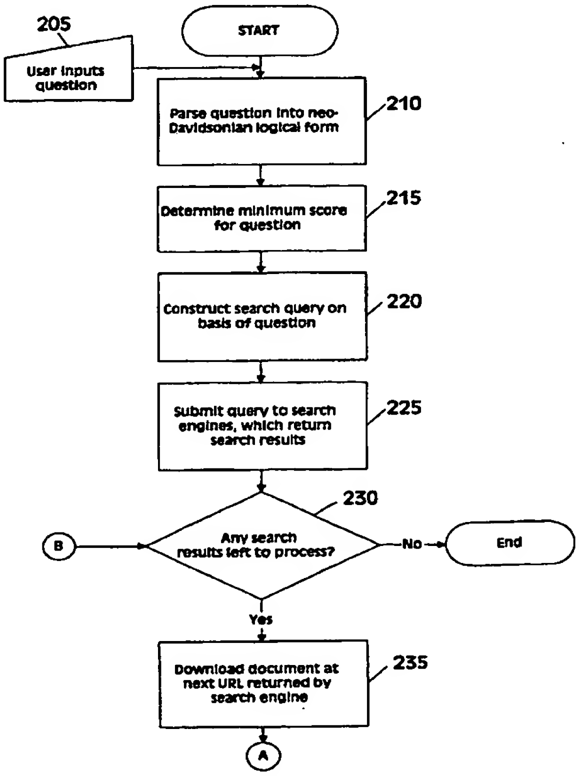
(81) Designated States: AT, AU, BG, BR, CA, CH, CN, CZ, DE, DK, ES, FI, GB, HU, IL, IS, JP, KR, LC, LU, MK, MX, NO, NZ, PL, PT, RO, RU, SE, SG, SI, SK, VN, YU, European patent (AT, BE, CH, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

Published
With international search report.
Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

(54) Title: METHOD AND APPARATUS FOR NATURAL LANGUAGE QUERYING AND SEMANTIC SEARCHING OF AN INFORMATION DATABASE

(57) Abstract

The invention relates to methods and apparatuses for receiving a user's query in natural language (e.g., English) form, searching an electronic database for sentences that may provide semantically meaningful answers to the query, identifying those sentences that are deemed to answer the question, and quantifying the degree to which the sentences answer the question.


205
User inputs question
START
Parse question into neo-Davidsonian logical form
210
Determine minimum score for question
215
Construct search query on basis of question
220
Submit query to search engines, which return search results
225
230
B
Any search results left to process?
No
End
Yes
Download document at next URL returned by search engine
235
A

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania
AM	Armenia
AT	Austria
AU	Australia
AZ	Azerbaijan
BA	Bosnia and Herzegovina
BB	Barbados
BE	Belgium
BF	Burkina Faso
BG	Bulgaria
BJ	Benin
BR	Brazil
BY	Belarus
CA	Canada
CF	Central African Republic
CG	Congo
CH	Switzerland
CI	Côte d'Ivoire
CM	Cameroon
CN	China
CU	Cuba
CZ	Czech Republic
DE	Germany
DK	Denmark
EE	Estonia
ES	Spain
FI	Finland
FR	France
GA	Gabon
GB	United Kingdom
GE	Georgia
GH	Ghana
GN	Guinea
GR	Greece
HU	Hungary
IE	Ireland
IL	Israel
IS	Iceland
IT	Italy
JP	Japan
KE	Kenya
KG	Kyrgyzstan
KP	Democratic People's Republic of Korea
KR	Republic of Korea
KZ	Kazakstan
LC	Saint Lucia
LI	Liechtenstein
LK	Sri Lanka
LR	Liberia
LS	Lesotho
LT	Lithuania
LU	Luxembourg
LV	Latvia
MC	Monaco
MD	Republic of Moldova
MG	Madagascar
MK	The former Yugoslav Republic of Macedonia
ML	Mali
MN	Mongolia
MR	Mauritania
MW	Malawi
MX	Mexico
NE	Niger
NL	Netherlands
NO	Norway
NZ	New Zealand
PL	Poland
PT	Portugal
RO	Romania
RU	Russian Federation
SD	Sudan
SE	Sweden
SG	Singapore
SI	Slovenia
SK	Slovakia
SN	Senegal
SZ	Swaziland
TD	Chad
TG	Togo
TJ	Tajikistan
TM	Turkmenistan
TR	Turkey
TT	Trinidad and Tobago
UA	Ukraine
UG	Uganda
US	United States of America
UZ	Uzbekistan
VN	Viet Nam
YU	Yugoslavia
ZW	Zimbabwe

METHOD AND APPARATUS FOR NATURAL LANGUAGE QUERYING AND SEMANTIC SEARCHING OF AN INFORMATION DATABASE

FIELD OF THE INVENTION

This invention relates generally to the computerized searching of information databases and, more specifically, to searching the World Wide Web for answers to a natural language question.

BACKGROUND OF THE INVENTION

More and more information of the world's information is currently available on-line in the form of raw or loosely-formatted text. The operations and activities of the world's organizations and institutions, large and small, formal and informal, are increasingly stored in on-line repositories of memoranda, letters, text transcriptions, reports, catechisms of "Frequently Asked Questions" (FAQs), electronic mail, announcements, on-line newsgroup and bulletin board postings, World Wide Web homepages, catalogs, and brochures, and so on. Further, with the advent of Internet protocols for electronic exchanges of information, more and more of these documents are accessible to others throughout the world, at any time, and from any location.

The scale of this information collection presents a problem to those who would access it. There is no master catalogue of this material. Further, while attempts have been made to promulgate standards for content identification, these have not been widely adopted. With little indication of where information is contained in this multitude of electronic documents, how is the user to find the information that is desired?

The field of information retrieval has been addressing itself to this problem since the middle of this century. A number of standard approaches have been developed and refined in the last quarter-century. The most popular of these are *keyword-based methods*.

The simplest keyword-based method is keyword indexing. The method begins by representing a document as a collection of words or strings of symbols rather than as an ordered sequence of meaningful propositions. Sophisticated techniques are then employed to match a query for information, represented as a set of strings of letters, to documents, again represented as sets of strings of letters.

In a popular variation of keyword indexing, words in a document collection are indexed to the documents that contain them. In order to access useful documents, users present a system with some collection of keywords and are returned references or pointers to documents that contain those keywords. More sophisticated variants of this technique allow users to specify further constraints on relevant documents beyond containing at least one of the keywords listed. For example, the user may specify keywords that must not appear in returned documents, the proximity of keywords within a document, the presence of multi-keyword phrases, and other Boolean conditions on the words that are contained in a relevant document. Such Boolean keyword searching techniques allow the user to make sophisticated constraints on documents that are to be considered relevant to the query. These techniques increase the precision of the returned document set by allowing the user to request a more-focused set of documents from the information retrieval system.

A different class of keyword-based techniques automatically expands the user's query for documents to include equivalent variants of the user's query in order to increase the number of documents returned. This results in an increase in the information retrieval system's *recall*. Examples of these techniques include "stemming" query terms to a root form so that all of the morphological variants of a keyword are matched in a document. For example, stemming "computation" and "computer" to the same root will return documents containing either term when queried with one of them. Another technique involves adding synonyms of query terms to the query so that, for example, a query on "3M" automatically returns documents that contain the string "Minnesota Mining and Manufacturing," pre-determined to be equivalent to "3M," as well.

The drawbacks of keyword-based techniques are well-known. There is a considerable computational and storage overhead associated with setting up the index of keywords for a collection of documents. These indices must change as the collection changes in order to be maximally accurate. Additionally, although precision-enhancing techniques allow the user to put sophisticated Boolean constraints on the documents returned, they burden the user with formulating a request in a special, unintuitive formalism in order to achieve that precision. Field experience has shown that users are intimidated by Boolean logic and reluctant to learn the formalism well enough to be able to make sufficiently precise queries. Recall-enhancing techniques, on the other hand, aim at increasing the number of documents returned per query.

While this increases the chance that all of the relevant documents will be returned, it dramatically increases the amount of time the user must spend on an information retrieval task, if the user is to survey every returned document for the required information. For many tasks, the amount of information returned will be too large for the user to survey completely, and the user will be likely to guess at which of the returned documents should receive attention.

More importantly, there is an inherent limitation in keyword-based techniques because they represent documents only as collections of words rather than as meaningful expressions arranged into text for some communicative purpose. A sentence is more than a set of words; the structure of the sentence does most of the work in determining the meaning of the sentence. Both of the previous classes of techniques fundamentally represent documents as collections of alphanumeric characters, i.e. combinations of letters, and use a combination of the user's input and the system's design to return relevant documents on the basis of the words they contain. Unless precise information about word order is specified, they will, therefore, fail to distinguish between "the man bit the dog" and "the man was bitten by the dog."

An alternative class of information retrieval techniques, called *content markup*, addresses this issue by representing the meaning of a document rather than merely the words it contains. These techniques involve marking up stored text (or even non-textual data) with a representation of the meaning or content of the document in some formalism or other. As a simple example, an implementation of such techniques would be to provide a set of photographs with a set of keywords representing what the photographs depict. This would have to be done manually. With a collection of text documents, an implementation of these techniques might involve marking up documents pertaining to financial transactions, say, with some representation of who is buying, who is selling, what is sold, and for how much. Documents could then be retrieved on the basis of their marked up annotations alone (e.g., "What did Company X buy?") or by means of their annotations as well as by keyword indexing.

The content markup approach has the advantage of pointing the user towards documents, or sections of documents, on the basis of the semantic or prepositional content of the document or its sections, rather than on the basis of the words that the document contains. This is certainly an improvement over the keyword approach. Consider the task of finding needed information in a library. One normally doesn't approach this task primarily by means of an index of all the words in the all the books in the library. One uses a representation of the content of the books (a card catalog entry) to find the relevant books. Only once the relevant books are found does one use a keyword index (the index of a particular book) to find the relevant passages. On-line information retrieval, on the other hand, relies on a word-level representation of libraries of text to locate the information users want. There is no equivalent of a card catalog or book abstract available for on-line documents.

Content markup approaches can index a collection of information on the basis of the propositions that the text expresses or that characterize the text rather than the words that the text contains. In the example above, specific sorts of "metadata" (data about data) were attached to a collection of financial transaction documents, indicating who bought what from whom and for how much. This would provide a useful way to find all and only the documents that talk about acquisitions *by* Company X but not acquisitions *of* Company X. This distinction would be very difficult to represent within a keyword approach.

The obvious difficulty with the content markup approach, however, is the markup process itself. It is difficult to automate this process, because marking up the documents requires an understanding of a document or text. While it is easy to program a computer to index the words of a text; it has previously not been possible to program a computer to create a representation of the meaning of a text. Thus, content markup approaches have relied on manually created markup annotations of documents in a collection. This is a time-consuming, labor-intensive process that, again, must be constantly updated to keep in step with changes in the document collection.

Furthermore, content markups that seek to characterize the semantic content of an entire document or document section necessarily represent only some of the semantic content of that text. In our financial transactions example, any semantic content that is not represented in the metadata is not accessible to the information retrieval process. The unrepresented content will usually be most of the data; ultimately, the only complete representation of the semantic content of the document is the document itself.

Yet another class of applications, *semantic querying*, accepts a question submitted to the information retrieval system in a natural language format. A semantic analysis of the question is then used to translate the question into a specialized language or otherwise reformat it to aid in information retrieval. Some applications of this type may translate queries submitted in English into Boolean logic, or into specialized database query languages such as SQL. Others parse the question into a semantic representation that can be matched against marked up content. Others analyze the question and produce a set of synonymous or near-synonymous queries, hoping to increase the recall of the system.

This class of applications represents a new level of sophistication in that they attempt to automatically understand the request for the information. The goal of these systems is to make it easier for users to ask for information or to retrieve more information. They fall short of understanding the information they scan and return, however. These methods still treat the raw text information they process as sets of words, rather than meaningful texts. Thus, the semantic querying applications should properly be thought of as pre-processors to existing keyword-based or content markup techniques discussed previously.

The foregoing shows that there exists a need for an information retrieval technology that can generate semantic representations of both the question *and target text* on the fly and use these representations to allow the user to retrieve needed information. The Answer Me! system (the "invention") provides this new level of sophistication in information retrieval, constructing a semantic representation of retrieved text rather than just the question, in real time, to facilitate the retrieval of answers to questions posed in a natural language.

SUMMARY OF THE INVENTION

The invention is a software application designed to find answers to user-submitted queries posed in English from on-line documents. The invention consists of three major components (application programs) that will be described in detail below: a *user interface*, a *parser*, and a sentence *evaluator* that determines the extent to which a given sentence answers a submitted question.

The user begins the process of retrieving information by submitting a natural language questions (e.g., English) to the user interface. For example, the user might ask, "When was Pluto discovered?" The submitted questions should be a direct request for the information, rather than a request to find the information. That is, users tend to anthropomorphize systems, but they should not make an indirect request for the information, asking, for example, "Can you find me the date of Pluto's discovery?" The question is then parsed into a form that will be useful for comparison with the returned answers. In addition, a minimum score that returned answers must meet or exceed is determined.

Next, the user interface accesses a database to identify a set of relevant documents to process for answers. This candidate set of documents might consist of the entire collection of a user's email messages, for example. In other cases (e.g., for WWW documents), it will be more efficient to process only a subset of a document collection. In such cases, the submitted question can be mapped to a keyword index query in order to narrow the range of candidate documents that may contain an answer to the submitted question.

Having identified a set of candidate documents, the invention processes each document, sentence by sentence, looking for answers to the submitted question. For each sentence, a judgment is made whether or not to *parse* the sentence into its thematic representation. This decision is based on the presence or absence of keywords from the query in the sentence. Sentences with no keywords from the query are discarded, and those with keywords are kept for further processing.

Once a decision has been made to keep a sentence, a parse (a semantic as well as a syntactic representation) of the sentence is created. Each sentence is taken to represent an event or state, with each phrase within the sentence representing some role in that event or state. Thus, one phrase may represent the agents of an event, another the theme of the event (that which is acted upon), another the instrument, and so on. The result is a structure consisting of an event of a specified type, plus a series of relationships specifying exactly how the participants in the event participate in the event. For example, "Brutus stabbed Caesar" would be represented as expressing the existence of a stabbing event, with Brutus as the agent of the event, and Caesar as the "theme" (undergoer) of the event.

Thus, Answer Me! relies on detailed knowledge of the syntax and semantics of verbs (events) in contrast to other Artificial Intelligence systems that have been based on a detailed representation of the relationships of nouns (objects) -- indicating, for example, that a foot is a part of a leg, and so on. The class of English verbs is much smaller than the class of nouns, and they have a smaller range of meanings. Therefore, a detailed representation of the nature of objects requires much more storage space and is much more complex than a detailed representation of the nature of events.

Finally, the similarity of the semantic representation of the candidate sentence to the similarly parsed question is *evaluated*. To what extent do they represent similar events? If there is a close enough match, the sentence is returned as an answer to the submitted question. The above steps are repeated for all sentences in the document, and all documents in the set of candidate documents. A metric reflecting the scoring of each document is presented to the user and can be used to order the answers. The metric is derived with the aid of a data structure that represents the relationship of various events on the basis of the verbs that express them. The data structure divides thousands of English verbs into various semantic classes and subclasses, representing relationships of synonymy and near synonymy between verbs. A metric of the closeness of meaning between any two verbs can be determined on the basis of their relative relationship within the data structure. A hypertext link to the documents from which the answers came can also be provided.

The major advantages of this approach over the keyword-indexing, content markup, and query analysis approaches are obvious. First of all, since the invention is attuned to what a sentence says, rather than merely the keywords it contains, it can retrieve information much more precisely. Secondly, since the semantic analysis of the text occurs in real time and on the fly, no time- , memory- and labor- consuming markup is required. Thirdly, the semantic content of each sentence is represented as opposed to a compressed semantic characterization of an entire document or document section. Fourth, the invention contains knowledge about the similarity of word meanings. The invention contains a knowledge base of event types, and so, can recognize that a *hunting* event, for example, is semantically close to a *seeking* event, while recognizing the distinct syntactic characteristics of the verbs "hunt" and "seek." Lastly, and most importantly, by rapidly processing all of the sentences of a document semantically -- not just the submitted question -- the invention radically speeds up the process of finding answers to specific questions with a high degree of both precision and recall.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is a high level block diagram of a computer adapted to perform the method of the invention.

Figures 2A - 2B show a flow diagram of the major functions that combine to enable the method of the invention.

Figures 3A - 3C show a flow diagram indicating the functioning of the parser that lies at the heart of the invention.

Figures 4A - 4D show a representative response, to a user question, in the form of an automatically generated HTML page.

DETAILED DESCRIPTION

Answer Me! locates text relevant to a user-specified question posed in a natural language format; analyzes that text for sentences that may provide answers to the question; identifies those sentences that are deemed to answer the question; and evaluates the degree to which those sentences actually constitute answers to the question.

1. System Implementation

Figure 1 shows a block diagram of a computer system adapted to perform the method of the invention. Via a bus 105, the central processing unit ("CPU") 110 is connected to a random access memory 115, a user interface terminal 120, and a local storage 125. During operation, application programs (conceptually, a *user interface*, *parser* and *evaluator* -- described more fully below) are downloaded from local storage 125 to RAM 115 for execution by CPU 110. Local storage 125 may be a magnetic, optical, magneto-optical, electronic, or any other device capable of storing the necessary application programs. As a matter of convenience, the term "electronically accessible" shall be understood to encompass all such storage devices. The local storage 125 also stores the documents which are to be searched for the answer to the user's question. Of course, in a networked environment, documents could actually originate from a remote site, and the search resources used may be remote search resources as well. In that case, local storage 125 need not be a traditional mass storage device, but need only be a memory device of sufficient capacity to receive and buffer the desired information. Hence, either the local storage 125 or the remote site could serve as a document depository. In the preferred embodiment, the documents to be searched reside on the World Wide Web ("WWW") and may be accessed via an Internet browser (e.g., Netscape); currently, a stand-alone user interface allows the user to input questions and view the progress of the processing, but user interaction may be accomplished in a variety of ways. Of course, distributed computing technologies (e.g. Sun's Java applets, Next's Web objects or EÓlas' Weblets) also allow the remote location and/or execution of the application programs themselves. Thus, the great operational flexibility afforded by a networked computing enrollment allows the system to be deployed for client, server, or hybrid operation depending on the local or remote provisioning of the application programs and/or documents.

In an embodiment suitable for standard PC based environments, Answer Me! is implemented as follows:

- Hardware: 486 or higher-speed Intel processor.
- Operating system: Windows 95 or Windows NT Workstation or Server v.3.5.1.
- Memory: A minimum of 8 megabytes of RAM (16 megabytes recommended) and a minimum of 5 megabytes of free disk space.
- Network connection: TCP/IP Internet dialup or permanent network connection using a 32-bit TCP/IP Winsock layer.
- Web browser: Netscape Navigator or Microsoft Internet Explorer.
- Search resources: Digital Equipment Corporation's Alta Vista search engine and InfoSeek's FAQ search service.

In this embodiment, Answer Me! was written in Visual C++ and compiled using Microsoft's Visual C++ Developer Studio v. 4.2. The above-mentioned application programs are implemented in a binary executable file (answerer.exe, ~1.5 Mb) and five associated data files. These include a part of speech data file (lexicon.dat, ~250 KB), a verb classes index (evca93.txt, ~75KB), a thematic grid index (theta.txt, ~15 KB), and two initialization files (ansrme.ini, 1 KB, and wlres.ini, 2 KB).

2. Querying and Candidate Answer Retrieval

Referring now to Figure 2, the method of the invention will now be described with respect to the above example of a WWW searcher. However, those skilled in the art will appreciate that the invention is usable for electronic document searching generally. At step 205, the user inputs the question and, at step 210, the question is parsed into a form useful for comparison with the returned answers. This form, which is known to those skilled in the art as neo-Davidsonian logical form, is also used during parsing of the returned answers, and will be described in detail in the following section entitled "**Sentence Parsing**." At step 215, a minimum score requirement is calculated for the question, which any answer returned to the user must exceed. A detailed discussion of the minimum score calculation is deferred until the section entitled "**Evaluation**" where all of the invention's scoring functions are discussed in a unified manner. At step 220, an appropriate search-engine query (e.g., the traditional keyboard-based, content markup or query analysis technique described in the **BACKGROUND OF THE INVENTION**) is constructed on the basis of the question. At step 225, the search engine query is submitted to the search engine or engines (e.g., WWW sites such as Alta Vista or InfoSeek) with which the invention has been enabled to communicate. These search engines return a set of pointers, to text documents, consisting of the documents' Uniform Resource Locations ("URLs"), i.e., their addresses on the WWW. In this way, a *candidate* set of documents that *may* provide answers to the question is obtained. The invention will then use these addresses to retrieve each document, one by one, and process them to extract answers. In simpler contexts, e.g., where the searched database is highly specialized or relatively small, it may not be necessary to identify a subset of candidate documents and steps 220-225 can be omitted. At step 230, the process begins by looping through the list of returned search results. At step 235, each search result (a document at its returned URL) is downloaded to local storage 125 for linguistic processing. At step 240, the document is *tokenized* to yield a collection of sentences. During *tokenizing*, a document in a computer-readable format (e.g., a web page in HTML) is preprocessed for subsequent semantical analysis (parsing) by stripping out formatting tags (e.g., HTML tags) and other non-textual characters, while stepping through the document one character at a time until a sentence boundary is found. The tokenizer includes a routine for detecting abbreviations so that sentences are not ended prematurely.

3. Sentence Parsing

At step 245, now within a particular document, the process beings looping through each of the sentences. At step 250, each sentence is parsed as will be described in more detail with respect to Figures 3A-3D (collectively referred to as "Figure 3") below. The parsing of each sentence of Answer Me! comprises the following three steps:

a) Part of Speech Tagging

b) Phrase Identification

c) Linking of Phrases to Thematic Roles

Steps (a) and (b) may be thought of as grammatical (or syntactical) steps, while step (c) may be thought of as a semantic operation. Together, the three steps constitute what shall be referred to herein as semantic analysis, the term "semantic" as a matter of convenience being used throughout this document to include both semantic and syntactic operations.

The outcome of the parser is a mapping of each sentence (whatever its mood -- indicative, interrogative, or imperative) into a representation of its *logical* or *semantic* form. The semantic formalism used is a member of the family known to those skilled in the art as *neo-Davidsonian logical form*.

In his 1967 paper, "The Logical Form of Action Sentences" (reprinted in D. Davidson, *Essays on Actions and Events*, Oxford: Clarendon Press, 1980), Donald Davidson theorized that the logical form of at least some natural language sentences involves first-order quantification over a covert event position. Davidson offered this hypothesis primarily in order to explain "adverb-dropping inferences" in a natural way and, more importantly, using a finite logical vocabulary. Thus, in order to explain how it is that "John buttered the toast at midnight" entails "John buttered the toast", Davidson proposed that the logical form of the first sentence is actually that of a conjunction of predicates with an event argument bound by existential quantification over an event (here, "buttering"). In first-order logical notation, the truth conditions of the sentence "John buttered the toast at midnight" would thus be (ignoring tense):

Ee(buttering(e,John,toast) & at(e,midnight)) (1)

where "Ee" represents an existential quantifier binding an *event* variable *e*. Informally, this notation states that there is an event e, that the event is an *argument* of the predicate *butter* along with "John" and "the toast," and that the event was at midnight. Thus, the fact that (1) entails "John buttered the toast" can be explained straightforwardly as an instance of conjunction elimination: from "A and B" infer "A" (for sentences A and B). Davidson's theory eliminated the need for an infinitely large vocabulary of n-place predicates, one for each possible set of adjuncts modifying a verb, and a set of inference rules relating them. Thus, in the above example, an indefinitely large set of predicates -- *butter* (for John buttered the toast), *butter-at* (for John buttered the toast at midnight), *butter-in* (for John buttered the toast in the bathroom), *butter-at-in* (for John buttered the toast at midnight in the bathroom) and so on -- is eliminated in favor of the single predicate *butter*.

Davidson's theory implicitly distinguished a verb's *arguments*, or the minimal set of expressions a verb requires in order to form a grammatical sentence, from what linguistic theory terms its *adjuncts*, or those expressions that may be added to the arguments of a sentence to express the time or place at which an event took place. In the example above, "John" and "the toast" are arguments of the verb "to butter" since the verb requires both a subject and a direct object noun phrase in order to form a grammatical sentence; the phrases "at midnight" and "in the bathroom" are considered adjuncts of the verb "to butter" since prepositional phrases of these types are not necessary to form a grammatical sentence with "to butter" as the main verb.

Neo-Davidsonian analyses of the logical form of sentences go one step further than Davidsonian analyses in treating arguments and adjuncts equally as conjuncts existentially bound to the same event variable. Arguments are analyzed as bearing a particular *thematic (or "theta") role* within an event. Alternatively, the argument is sometimes said to bear a *thematic relation* to an event. Thus, the sentence "John buttered the toast" would be assigned the logical form:

Ee(buttering(e) & Agent(e, John) & Theme(e, the toast)) (2)

Informally, (2) states that there is an event e, that the event is a buttering, that the *agent* of the event is John, and that the *theme* of the event is the toast. Here *agent* and *theme* denote thematic roles. Thematic roles are gross distinctions among the ways in which things participate in events and states. The agent of an event is the participant who intentionally brings about the event. Necessary and sufficient conditions for being the theme of an event are the subject of considerable controversy in the literature, but, generally, the participant that undergoes the event, or that the event happens to, is the theme. A list of the thematic roles used within the analysis of the invention along with illustrative examples are given below:

External Argument (extarg): **John** is sad; John appointed **Mary** president

Predicate (pred): John is **a barber**; John appointed Mary **president**

Agent (ag): **John** gave the book to Mary

Source (source): John took the book **from Mary**

Goal (goal): John gave the book **to Mary**

Theme (th): John gave **the book** to Mary

Benefactive (ben): John threw a party **for Mary;** John threw **Mary** a party

Instrument (ins):John cut the cake **with a knife; the knife** cut the cake easily

Location (loc): John put the book **on the table**; John ate breakfast **before work**

Path (path): The planet circles **around the sun**

Manner (manner): The bread cuts **easily;** John cooked the spaghetti **by boiling it**

Purpose/Reason (purp): John skipped school **to see the game.**

Measure (measure): It snowed **a foot**, John weighed **200 pounds**

Result (result): John hammered the metal **flat**

Possessor (poss): **John** has the flu

Possessed (posd): John has **the flu**

Duration (dur): John drank beer **for an hour**

Conative (con): John poked **at the log**

Thematic roles are to be distinguished from grammatical roles such as *subject* and *direct object*, which refer to a phrase's position, rather than the way in which its referent participates in an event. Grammatical roles distinguish the syntactic position of a phrase in a sentence in relation to the sentence's verb or some other phrase-heading element. Two phrases may have different grammatical roles, but bear the same thematic relation as in the sentences:

John gave Mary a book (3)

John gave a book to Mary (4)

In sentence (3), "Mary" is the direct object, however, and in sentence (4) "a book" is the direct object. In both cases, however, John is the agent of the giving event, Mary is the goal (the thing to which the giving is directed), and the book is the theme, the thing that undergoes the giving. In logical notation:

Ee(giving(e)&Agent(e,John)&Theme(e, the book)&Goal(e, Mary)) (5)

Thus, on the neo-Davidsonian account, sentence (3) logically entails sentence (4), and vice versa. Unlike the neo-Davidsonian analysis, Davidsonian analyses cannot account for these entailments directly: in order to explain an inference from (3) to (4) or from (4) to (3), additional inference rules would be necessary in a Davidsonian theory. This is a clear advantage for neo-Davidsonian accounts since most verbs are like "give" in that they allow their arguments to embody a variety of grammatical and thematic roles.

The essential features of a neo-Davidsonian account of logical form are (i) quantification over implicit event variables, (ii) thematic role analysis of the semantics of argument positions, and (iii) the treatment of verbs as one-place predicates of events. The invention thus embodies a neo-Davidsonian analysis of sentences. The analysis differs from discussions of neo-Davidsonian logical form in the literature only in how these analyses are encoded. Rather than embody the logical forms as formulas of first-order logic, the invention maps parsed sentences into C++ data structures (called "objects") stored in computer memory. The analysis could equally well map sentences into other computational data structures, such as assertions in Prolog or Java classes. In the preferred embodiment, each C++ object mapped to a sentence represents an event; the type of event and the thematic roles identifying the participants in the event are given as member variables within that object.

Neo-Davidsonian logical form is described in more detail in the following references: Terence Parsons, *Events in the Semantics of English: A Study in Subatomic Semantics*, MIT Press (1990); Gabriel Segal and Richard Larson, *Knowledge of Meaning: An Introduction to Semantic Theory*, MIT Press (1995); James Higginbotham, "On Semantics," *Linguistic Inquiry* 16:547-593 (1983). Further references on linking theory include: Edwin Williams, *Thematic Structure in Syntax*, MIT Press (1994); David Pesetsky, *Zero Syntax: Experiencers and Cascades*, MIT Press (1995). The nature and status of thematic roles in contemporary semantic theory is described more fully in the following references: William Frawley, *Linguistic Semantics*, Lawrence Erlbaum Associates (1992), Chapter 5; David Dowty, "On the Semantic Content of the Notion of

'Thematic Role'," in G. Chierchia, B. Partee and R. Turner, eds., *Properties, Types and Meaning*, vol. 2, *Semantic Issues* (pp. 69-129) Kluwer (1989); "Thematic Proto-Roles and Argument Selection," *Language* 67:547-619 (1991); Terence Parsons, "Thematic Relations and Arguments," *Linguistic Inquiry* 26:635-662 (1995).

The invention relies on and develops recent theoretical work on the relationship of grammatical roles and thematic roles in the literature of the rubrics of neo-Davidsonian logical form and linking theory, as described in Brian Edward Ulicny, *Issues in the Philosophical Foundations of Lexical Semantics*, Chapter 3, Doctoral Dissertation, Department of Linguistics and Philosophy, Massachusetts Institute of Technology (accepted May, 1993) and Douglas A. Jones, Robert C. Benwick, Franklin Cho, Zeeshan R. Khan, Naoyuki Nomura, Anand Radharkrishnan, Ulrich Sauerland and Brian Ulicny, *Verb Classes and Alternations in Bangla, German, English and Korean*, Memo 1517, Artificial Intelligence Laboratory, Massachusetts Institute of Technology (dated August, 1994; available for distribution as an AI Lab Memo in Spring, 1996). The AI Lab Memo describes a crude parser that implemented a function from sentences to a *grammaticality judgment* of good or bad. A sentence was deemed good (grammatical) if it had at least one acceptable parse and was bad otherwise. Thus, the AI Lab Memo parser did not attempt to find a unique parse for a sentence, and its resulting logical form analysis would effectively reflect several thematic assignments. In addition, it had a very limited vocabulary and did no morphological analysis. Its coverage was extremely limited, being basically designed to analyze the example sentences of Beth Levin's book, *English Verb Classes* *and Alternations*. Thus, only past tense verb forms were handled and no attempt was made to handle intra-sentence punctuation, questions, imperatives, relative clauses, passive verbs, pronoun resolution, or phrase movement of any sort. Finally, the AI Lab Memo parser operated on single sentences rather than entire documents, and did not attempt to evaluate sentences as answers to submitted questions.

In contrast, the parser of the present invention represents a significant improvement over the AI Lab Memo parser's deficiencies in each of the above-mentioned semantical and operational respects. Referring now to Figure 3, the present invention's process of parsing a sentence summarized as step 250 in Figure 2 is explained in greater detail.

a) Part of Speech Tagging

At steps 305 and 310, the sentence in the buffer is passed, one word at a time, to a part of speech tagger for morphological analysis. Morphological variants of words result, for example, from situations such as prefix/suffix addition, inflection for past tense, etc. At step 315, the part of speech tagger assigns a part of speech tag to each word of the sentence based upon the word's context and occurrence. The part of speech tag is selected from a total of 48 predetermined choices, such as: noun, verb, preposition, adjective, etc. The actual tag assigned depends at least in part on the last tag assigned, which reflects the selection properties of the preceding word. For example, if a word can be both a noun and a verb (e.g. *book*), the tagger will return NOUN if the preceding word was a determiner (as in *the book*) or VERB if the preceding word was an auxiliary (as in *might book the hotel room*). Reference to the context of a word in tagging is especially crucial for English, which has an inordinate number of verbs that are homonymous with nouns and adjectives that are homonymous with verbs; morphology won't distinguish the part-of-speech in these cases.

b) Phrase Identification

At step 320, the sentence is analyzed in accordance with the syntactic theory known to those skilled in the art as *Case Theory*. Case Theory, which is a subtheory of the school of syntactic analysis known as Government and Binding Theory, asserts that every phrase of a sentence must paired one-to-one with a *case assigner*. Parts of speech that are case assigners include: TENSE (INFLECTION), VERBS and PREPOSITIONS. Thus, the sentence

John hit the ball Mary

(6)

is ungrammatical, in part, because the phrase "Mary" has nothing to assign it Case. (Case need not be overtly marked in the morphology of a word.) In the context of the invention, case markers are used to identify *phrase boundaries* and to parse sentences into their constituent phrases. These and other aspects of Case Theory are described more fully in Liliane Haegeman, *Introduction to Government and Binding Theory, 2nd Edition*, Basil Blackwell (1994), which is incorporated herein by reference. Thus, at step 325, the sentence is divided into its constituent phrases by successively adding its words to a phrase buffer until, at step 330, the presence of a new case assigner is encountered (if overt) or can be inferred (if covert). For example, the sentence "Brutus stabbed Caesar in the forum" will be divided into the phrases: <Brutus><stabbed><Caesar><in the forum> on the basis of encountering the case assigners Past (TENSE), "stab" (VERB), and "in" (PREPOSITION).

At step 330, when an entire phrase has been detected, that phrase is assigned a *feature* that characterizes the phrase as a whole or that the phrase may be said to project. A noun phrase is assigned the (default) feature *N*. A prepositional phrase's head preposition is assigned as its phrase feature; the properties of this preposition will determine the role it plays in the sentence. In other cases, features are assigned to a phrase on the basis of linguistic rules. For example, in the sentence "John cooked the spaghetti by boiling it", the phrase "by boiling it" would be assigned the feature MANNER because it describes how the event described by the main verb was accomplished.

Verbs are inserted into the sentence's phrasal constituents and assigned the feature *V*. Only the head verb is included; auxiliaries, negations and other elements that modify the head verb are not included. Furthermore, a verb's TENSE and ASPECT are not considered for the purposes of information retrieval here. By way of example, the sentence "Brutus killed Caesar in the forum by stabbing him" would be mapped to the phrasal representation <N, Brutus><V, kill><N, Caesar><IN, in the forum><MANNER, by stabbing him>.

There are cases in which certain adjustments are made by the invention - for example, in the case of interrogative or passive sentences -- to counter the effects of what those skilled in the art of Government and Binding Theory consider the movement of phrases from their default position. This is done in preparation for the thematic linking step, so that all arguments of a verb will appear in a canonical position with respect to the verb of which they are an argument. The surface word order of questions is understood in Government-Binding syntax to result from the movement of underlying phrases from their position in what is called D-structure through the movement of various phrases and elements to the front of the sentence. ' Thus,

Who did Brutus stab?
(7)

is understood to result from the movement of "who" and the past tense morpheme from their (default) position as the tense node and direct object in the sentence, as in "Brutus stab+PAST (stabbed) who". The invention thus analyzes the question (7) as the sequence of phrases: <N,Brutus><V,stab><N,who> rather than <N,who><N,Brutus><V,stab> by reversing the process of question-formation in English. The auxiliary "did" shows up in the question (7) as the realization of the past tense marker formerly visible in the morphology of the verb form "stabbed". Because "did" does not function as a verb in (7) (unlike in "Brutus did a dance"), it is ignored. Similar movement and reanalaysis by the invention also occurs in relative clause constructions and raising verb constructions (transforming, e.g., "It is easy to please John" to "John is easy to please").

Other forms of adjustments may also be performed at this stage, if necessary. For example, Answer Me! replaces pronouns with their antecedents when such replacements are unambiguous. Thus, the sentence "John shaved himself" is mapped to the set of phrases: <N, John> <V, shave> <N, John>. Similarly, "John met the man who rescued his dog" is mapped to: <N, John> <V, met> <N, the man> <N, the man> <V, rescue> <N, his dog>. While "his dog" cannot mean the rescuer's dog, it can mean someone other than John's dog; thus, no attempt is made to link the potentially ambiguous "his" with its antecedent.

c) Linking of Phrases to Thematic Roles

At this point, the sentence has been grammatically segmented into a collection of phrases. These phrases fall into one of the following categories: *verbs, arguments* of those verbs, or *adjunct phrases* -- optional phrases indicating, for example, the location or time of an event. The last step in parsing is to *link* each verb with its associated arguments using the technique of *thematic analysis.* Thematic roles are also assigned to adjunct phrases when this is possible.

The invention contains a data structure that associates each verb with the set of "thematic grids" it can select as arguments. A thematic grid is a vector of thematic roles. Since verbs may assume several forms, based on their inflection for tense and agreement, the index is based on a stemmed form of the verb. The stemmed form of the verb is derived by means of an algorithm known to those skilled in the art as the Porter stemming algorithm, although other well-known stemming techniques would work equally well. The Porter stemming algorithm is described more fully in Edward Frakes and Ricardo Baeza-Yates, *Information Retrieval: Data Structures and Algorithms*, Chapter 8, "Stemming Algorithms," Englewood Cliffs: Prentice-Hall (1992), which is incorporated herein by reference. Thus, the verb forms "give", "gives", "given" and so on, are mapped to the stem "giv" by the Porter stemming algorithm. This stemmed form is mapped to a set of thematic grids, containing <agent, theme, goal> corresponding to "John gave a book to Mary", <agent, N/goal, theme> corresponding to "John gave Mary a book" and so on.

In addition to those mentioned above, certain specialized conditions on thematic roles may be used by the linking algorithm for greater specificity. One such specialized thematic role involving a slash (i.e., A/B) requires two conditions on a phrase: the phrase must be headed by the feature compatible with the element to left of the slash (A), and the phrase must play the role to the right of the slash (B). This is useful for verbs that select particular prepositions in certain contexts, or for thematic roles which are usually headed by prepositions but sometimes appear as plain noun phrases, as with "N/goal" above. For example, in the sentence "John sprayed the wall with paint." the verb "spray" is associated with thematic grid <agent, N/location, with/theme>, among others. To discharge the thematic role "with/theme", a phrase headed with the preposition "with" must be the second argument to the right of the verb when it is linked. It will be assigned the thematic role Theme.

Another specialized thematic role "V/theta" (for some thematic role theta) is used to indicate the presence of verbs that *incorporate* nouns playing a thematic role. This analysis derives from theoretical work by Mark Baker; see his monograph *Incorporation: A Theory of Grammatical Function Changing,* University of Chicago Press, 1988, for further details on the conditions under which verbs are thought to incorporate arguments. As a simple example, in the sentence "It snowed," the analysis of the invention takes the verb "snow" to have incorporated the noun "snow", which is assigned the Theme role of the event.

Referring specifically to Figure 3, the actual linking of a verb to its corresponding thematic roles is summarized as a single step 335. However, this is really a two-part step: (i) identification of all possible (candidate) thematic grids for the verb, and (ii) selection of a particular (best) thematic grid for the verb from the list of possible candidates.

(i) Determination of Candidate Thematic Grids

Identification of candidate thematic grids involves the use of two indices. The first index is a classification of all possible verbs by verb classes. The second index is a listing of all possible thematic roles selected by each verb class of the first index. These two indices are described in greater detail in the paragraphs below.

Every verb in the verb class index is assigned to some class (or classes) based on the verb's meaning and the syntactic behavior of the verb's arguments. For details of the classification scheme, see Beth Levin, *English Verb Classes and Alternations: A Preliminary Investigation*, University of Chicago Press (1993), which is incorporated herein by reference. The set of syntactic frames within which a verb can express its arguments is known to those skilled in the art as its set of *alternations*. For example, verbs of the *Poke* class participate in the following alternations (* denotes ungrammatical sentences, put here for illustration):

"Allison poked the needle through/into the cloth."
"Allison poked the needle against the cloth."
"Allison poked the cloth with a needle."
"The needle poked the cloth."
"Allison poked the cloth."
* "The cloth poked."
"Allison poked Daisy in the ribs."
* "The cloth pokes easily."
"Allison poked Daisy's ribs."
*"Allison poked the ribs (meaning Daisy's ribs)."

Members of this class include the verbs *dig, jab, pierce, poke, prick,* and *stick*. Their behavior contrasts with those of the *Touch* class, for instance, which does not allow *through* or *into* phrases as arguments. That is, the sentence "Carrie touched the stick through/into the cat" is ungrammatical. This would seem to indicate that proper usage of the Poke verbs necessarily involves some sort of directed motion, which can be expressed by a *through* or *into* phrase, whereas the Touch verbs do not. The Touch verbs simply express a relationship between two objects, while the Poke verbs specify something about the relationship of the instrument used in the poking to the material substance of the thing poked.

The second index maps a verb class to the *thematic grids* associated with the various alternations in which the verbs in that verb class may participate. For example, four of the thematic grids that *clear* (and other verbs of its class) project are:

<Agent, Theme, Location> as in "John cleared the dishes from the table."

<Agent, Location, of/Theme> as in "John cleared the table of dishes."

<Theme> as in "The screen cleared."

<Agent, Theme> as in "John cleared the screen."

In the method of the invention, each verb in the sentence is looked up in the first index to yield one or more verb classes. Then, for each verb class, all its possible thematic grids are determined from the second index. In this way, all possible candidate thematic grids for a given verb are determined.

At step 340, any necessary adjustments to the thematic structure of *passivized verbs* will be made. Passivized verbs do not deploy the thematic role of their subject (except, optionally, in a *by*-phrase); thus, a thematic role for the non-passivized verb's subject should be assigned only if there is a *by*-phrase (e.g., "Brutus was stabbed by Caesar.").

(ii) Selecting the Best Thematic Grid for a Verb

Having segmented a sentence into phrases and determined the verbs' candidate thematic grids, the best candidate thematic grid must be determined for each verb and its associated arguments. At step 345, the candidate thematic grids are rearranged in order of decreasing length. This allows the longest grids to be evaluated first, because a successful match to a longer (and thus more precise) grid is preferable to a shorter grid. At step 350, the linker steps through the thematic grids one at a time and, at step 360, attempts to assign each verb and its arguments to a best thematic grid based on the verb's semantics. That is, the linker tries assigning thematic roles to phrases, starting with the leftmost verb in the sentence. If the thematic roles in the verb's thematic grid (e.g. <Agent, Theme>) are compatible with the features of the phrases immediately to the left of the verb and following the verb, then the phrases are assigned to those roles. For example, if a verb selects an Agent as its subject, or external argument, the linker will consider a phrase immediately to the left of the verb to be compatible if it is a noun phrase (has feature N) or has some other feature compatible with the Agent role.

One particular situation involving Agent roles deserves special consideration. Referential dependencies between pronouns and overt noun phrases within a sentence or due to ellipses, both of which might result in a multiplicity of phrases assigned to certain thematic roles, are handled by allowing only one set of thematic roles to be assigned. For example, if more than one Agent is found in a sentence, the second agent is appended to the previously assigned Agent phrase. This allows the system to recognize some anaphoric relations that might otherwise be missed. For example, the sentence "If a man owns a donkey, he beats it," contains two Agents ("a man," "he") and two Themes ("a donkey," "it"). By appending phrases to already assigned thematic roles, the system will be able to recognize the donkey sentence as an answer to the question "Does a man beat a donkey?"

Within the neo-Davidsonian paradigm, the invention's treatment of multiple verbs can be thought of as an endorsement of the axiom that for every group of basic level events, there is an event (a super-event) that consists of the occurrence of all those events. The agents of those constituent events are the agents of the super event; the themes of the constituent events are the themes of the super event, and so on.

At step 365, if all of the phrases are assigned a thematic role and all of the thematic grid's thematic roles have been assigned, we say that the linking has *converged;* otherwise it has *crashed.* At step 370, the first assignment of thematic roles that converges, if one occurs, is retained. Otherwise, at step 375, the assignment of thematic roles to phrases that comes closest to converging is retained. That is, the assignment that crashes the least badly is retained if none has converged. Phrases that don't get assigned a thematic role are assigned a special "adjunct" role. If convergence has not occurred, the process of steps 360-370 is repeated for the remaining candidate thematic grids, in an attempt to find a candidate grid that actually converges. Finally, at step 380, the best thematic grid is outputted for *score evaluation* (steps 255-260 of Figure 2).

Although the foregoing has been described in the context of indicative sentences (statements), other types of sentences such as interrogatives (questions) and imperatives (commands) are parsed by means of the same algorithm. For questions, the parser takes into account the movement of *wh-*

phrases and produces logical forms thematically equivalent to the corresponding statement by reordering the question's interrogative (wh-) phrases and by deleting the auxiliary as described previously in the section entitled **"Phrase Identification."** For imperatives, which have no subjects, the missing subject is not assigned a thematic role, but the relevant adjustment is made in judging whether the assignment has converged.

4. Evaluation

Referring now to Figure 2, at step 255, once the best parse has been found for a sentence, it is evaluated as to the degree to which it answers the submitted question. Answerhood is measured by a graded relation between a sentence and a question. A sentence may either be a full answer, a partial answer, or a non-answer to a submitted question. Partial answers with scores greater than a predetermined minimum score for the question are returned as well as full answers.

As mentioned previously (step 210 of Figure 2), after a submitted question has been parsed (in an identical manner to that described for sentences), it is assigned a minimum score that must be met or exceeded by a sentence, if that sentence is to be deemed an answer to the question. This score is given by the number of thematic roles assigned to the question. Thus, the questions "Who discovered Pluto?" and "Did Tombaugh discover Pluto?" would both be assigned a minimum score of 2, because they both contain an Agent ("Who" and "Tombaugh", respectively) and a Theme ("Pluto"). However, the question "How did Tombaugh discover Pluto?" would be assigned a minimum score of 3, since it also contains the additional thematic role of Manner ("How").

At step 255, a sentence is evaluated with respect to the submitted question as follows. First, a comparator determines, for each thematic role in the sentence, whether the phrase assigned to that thematic role in the question (e.g., the question's Agent) literally (exactly) matches a substring of the phrase assigned to that thematic role in the sentence. For each such match, a scorer increases the score by one. A sentence having a score of zero is discarded. Next, the comparator and scorer determine whether any of the question's thematic roles and if so, are occupied by *wh*-phrases (e.g., interrogatives such as "who," "which," "why," "how," "when," "where," and "what") in the sentence and, if so, increases the score by one per occupancy. In all the above, when dealing with the special case of an imperative sentence, its lack of a subject is taken into account so as not to exclude it from consideration on that basis. Finally, verb-based comparisons are made between questions and each sentence. Any such match, either between verbs or between verb classes, also increases the score by one per match. More generally, by arranging the verb classes in the first index into a tree structure, the semantic distance between the sentences and the question's verb classes could be quantified as the numerical distance between them in the tree. Such a numerical distance could be used an inverse measure of answerhood, with shorter distances causing larger score increases.

At step 260, sentences that score higher than the minimum score associated with the question are presented to the user, or otherwise stored, as answers to the submitted question. In the present invention, an HTML page with the answers, their scores, and a hypertext link to the page from which they were extracted, is constructed and automatically updated for the user, who accesses it through a Web browser.

Processing continues until all sentences (step 245) of all the documents returned by the search query (step 270) have been evaluated. Figures 4A-4D show a representative response to a user question of the form "When was Pluto discovered?" in the form of an automatically generated HTML page.

In the present invention, processing of the documents is done on an as-needed basis. The parses are not stored for future use. It would present no technical difficulty to store the data structures that result from the parsing in a database so that question-answering could directly access the stored parses, rather than parsing the sentences as needed.

Although the present invention has been described in terms of a particular embodiment, it will be appreciated that various modifications and alterations may be made without departing from the spirit and scope of the invention. Therefore, it is intended that the scope of the invention be limited only by the following claims.

CLAIMS

What is claimed is:

1. A method for searching an electronically accessible document depository to provide at least one answer to a question posed by a user in a natural language, comprising the steps of:

(a) identifying at least one candidate document, from said depository, that may contain at least one answer to said question;

(b) individually processing each said candidate document to locate said answers using thematic linguistic analysis; and

(c) presenting said answers to said user.

2. The method of claim 1 where the step of identifying said candidate document includes:

(a) receiving a natural language question;

(b) submitting an appropriate query, based on the question, to a search engine that references said depository; and

(c) receiving, from said search engine, an address for each said candidate document that may contain an answer to said question.

3. The method of claim 2 where the step of submitting said query includes formulating a keyword-based representation of the question to be submitted to the search engine.

4. The method of claim 2 where said document depository is the World Wide Web and where said candidate documents are web pages.

5. The method of claim 1 where the step of processing each said candidate document includes:

(a) identifying at least one sentence within said candidate document;

(b) determining whether each identified sentence should be parsed;

(c) for each sentence to be parsed, parsing it into its neo-Davidsonian logical form; and

(d) for each parsed sentence, evaluating the degree to which it constitutes an answer to said question.

6. The method of claim 5 where the step of parsing each said sentence includes:

(a) decomposing said sentence into at least one verb and a plurality of arguments associated with said verb, said verb and said arguments collectively defining a first plurality of phrases; and

(b) determining a thematic role corresponding to each said phrase.

7. The method of claim 6 further comprising the step of morphologically analyzing said sentence prior to said step of decomposing said sentence.

8. The method of claim 6 where said step of decomposing said sentence includes moving at least one phrase within said sentence.

9. The method of claim 8 where said step of moving said phrase is performed with respect to an interrogative sentence.

10. The method of claim 8 where said step of moving said phrase is performed with respect to a passive sentence.

11. The method of claim 8 where said step of moving said phrase is associated with a relative clause construction.

12. The method of claim 6 where said step of decomposing said sentence includes replacing at least one pronoun with an antecedent thereof.

13. The method of claim 6 where said step of determining said thematic roles includes:

(a) for each verb, determining at least one candidate thematic grid, each said candidate thematic grid including a plurality of thematic roles; and

(b) for each verb, selecting one of said candidate thematic grids whose thematic roles best match the arguments associated with that verb.

14. The method of claim 13 where said step of determining said thematic grid includes stemming a verb of said sentence.

15. The method of claim 5 where said sentence is of imperative form.

16. The method of claim 5 where said sentence is of interrogative form.

17. The method of claim 5 further comprising the step of parsing said question into a neo-Davidsonian logical form including a second plurality of phrases.

18. The method of claim 17 where said step of evaluating the degree to which said sentence constitutes an answer to said question includes:

(a) searching for matches between said sentence's first plurality of phrases and said question's second plurality of phrases; and

(b) incrementing a numerical score for said sentence upon the occurrence of each said match.

19. The method of claim 18 where said step of searching for matches includes literal matching.

20. The method of claim 18 where said step of searching for matches includes interrogative matching.

21. The method of claim 18 where said step of searching for matches includes verb-based matching.

22. The method of claim 18 where said step of presenting said answer to said user includes providing those sentences whose scores satisfy a minimum score requirement.

23. The method of claim 17 where said step of parsing said question includes:

(a) decomposing said question into at least one verb and a plurality of arguments associated with said verb, said verb and said arguments collectively defining said second plurality of phrases; and

(b) determining a thematic role corresponding to each said phrase.

24. The method of claim 23 where said step of determining said thematic roles includes:

(a) for each verb, determining at least one candidate thematic grid, each said candidate thematic grid including a plurality of thematic roles; and

(b) for each verb, selecting one of said candidate thematic grids whose thematic roles best match the arguments associated with that verb.

25. A system for searching an electronically accessible document depository to provide at least one answer to a question posed by a user in a natural language, comprising:

(a) a parser coupled to receive:

(i) said question, and

(ii) from said depository, at least a portion of said candidate document that may contain an answer to said question; and

(b) an evaluator coupled to receive from said parser:

(i) a first output derived from said question and

(ii) a second output derived from said portion of said candidate document; and

said evaluator configured to derive from said first and second outputs each said answer to be presented to said user.

26. The system of claim 25, further comprising a keyword-based searcher for identifying said candidate document from said depository.

27. The system of claim 25, where said depository is the World Wide Web and said candidate document is a web page.

28. The system of claim 25 further comprising a tokenizer coupled to said parser, said tokenizer configured to receive said candidate document from said depository and to identify said portion of said candidate document, said portion including at least one sentence of said candidate document.

29. The system of claim 28, where said evaluator includes:

(a) a comparator coupled to said parser to receive said first and second outputs and to determine any matches therebetween; and

(b) a scorer coupled to said comparator to receive said matches and to assign a rank to each said match.

30. The system of claim 29 where said matches and scores are literally-based.

31. The system of claim 29 where said matches and scores are interrogative-based.

32. The system of claim 29 where said matches and scores are verb-based.

33. The system of claim 25, where said parser includes

(a) a phrase boundary identifier configured to decompose:

(i) said portion of said candidate document into a first plurality of phrases, and

(ii) said question into a second plurality of phrases; and

(b) a linker coupled to said phrase boundary identifier, said linker configured to determine a thematic role corresponding to each of said phrases;

such that said first output includes said first plurality of phrases and their corresponding thematic roles and said second output includes said second plurality of phrases and their corresponding thematic roles.

Fig. 1

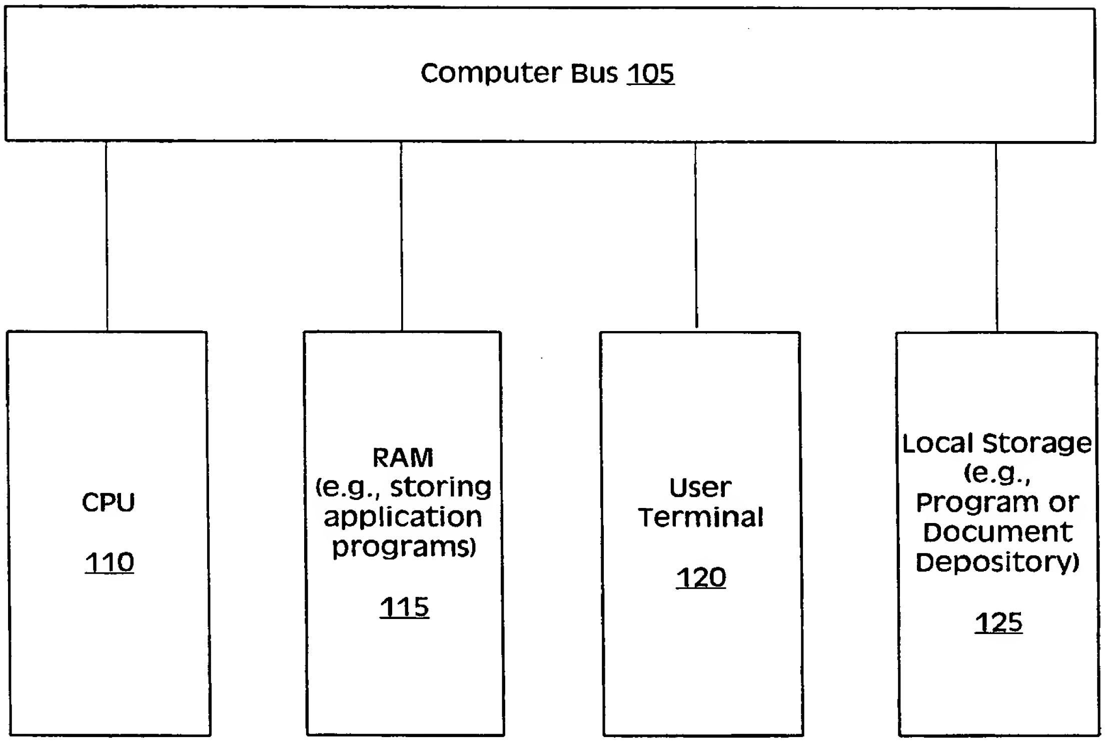

Computer Bus 105
CPU
110
RAM
(e.g., storing
application
programs)
115
User
Terminal
120
Local Storage
(e.g.,
Program or
Document
Depository)
125

Fig. 2A

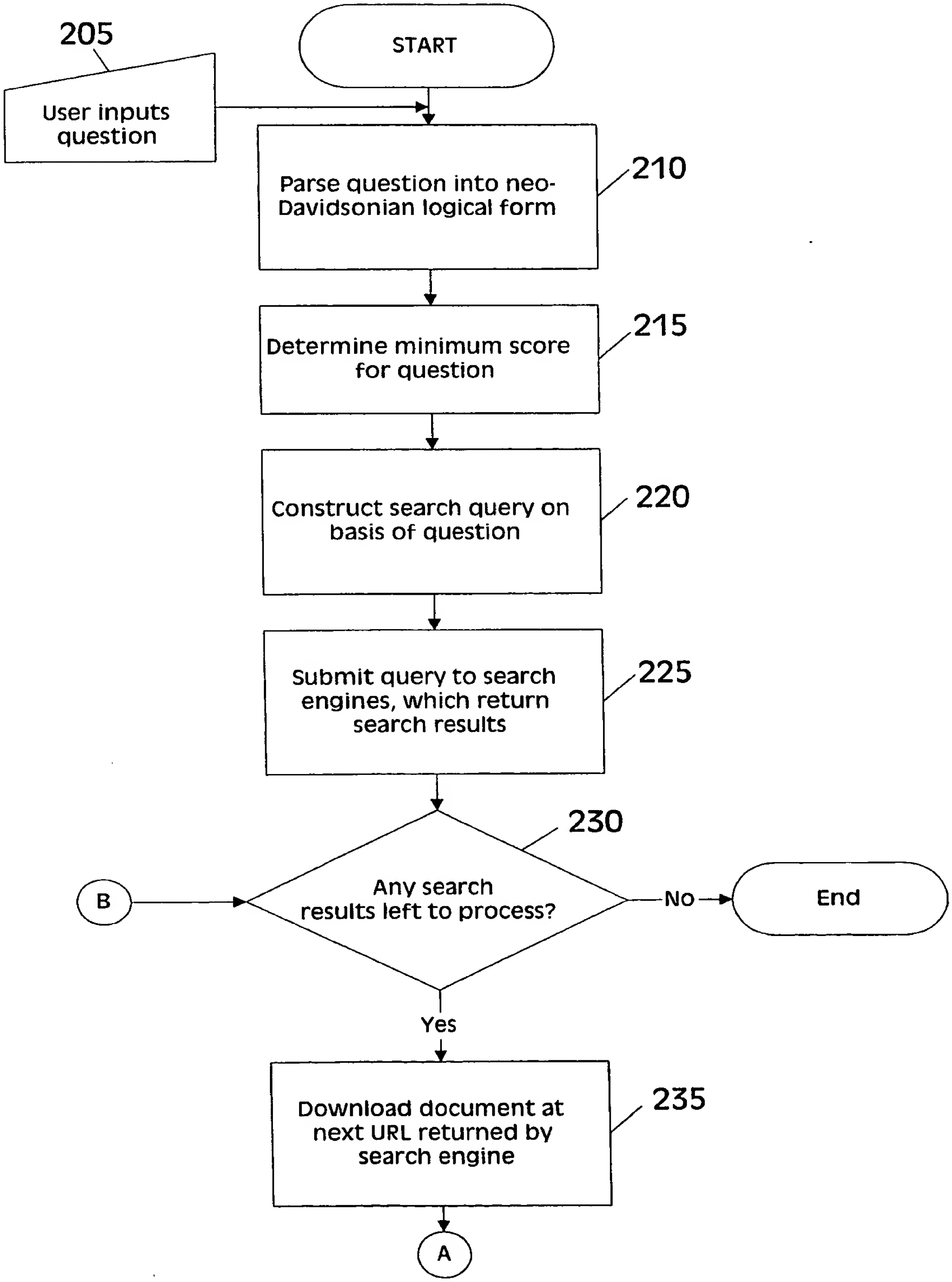

205
User inputs question
START
Parse question into neo-Davidsonian logical form
210
Determine minimum score for question
215
Construct search query on basis of question
220
Submit query to search engines, which return search results
225
230
B
Any search results left to process?
No
End
Yes
Download document at next URL returned by search engine
235
A

Fig. 2B

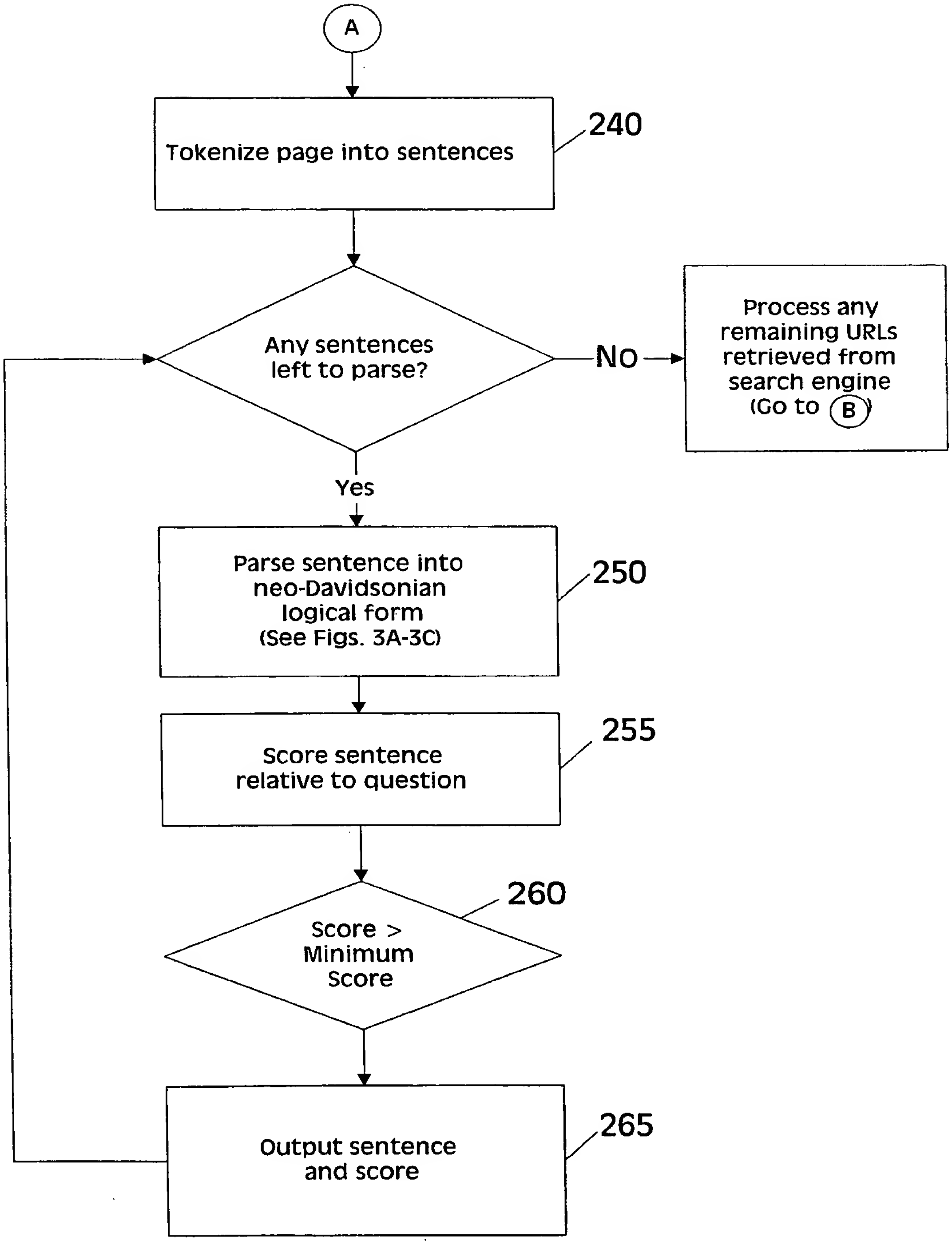

A
Tokenize page into sentences
240
Any sentences left to parse?
No
Process any remaining URLs retrieved from search engine (Go to B)
Yes
Parse sentence into neo-Davidsonian logical form (See Figs. 3A-3C)
250
Score sentence relative to question
255
Score > Minimum Score
260
Output sentence and score
265

Fig. 3A

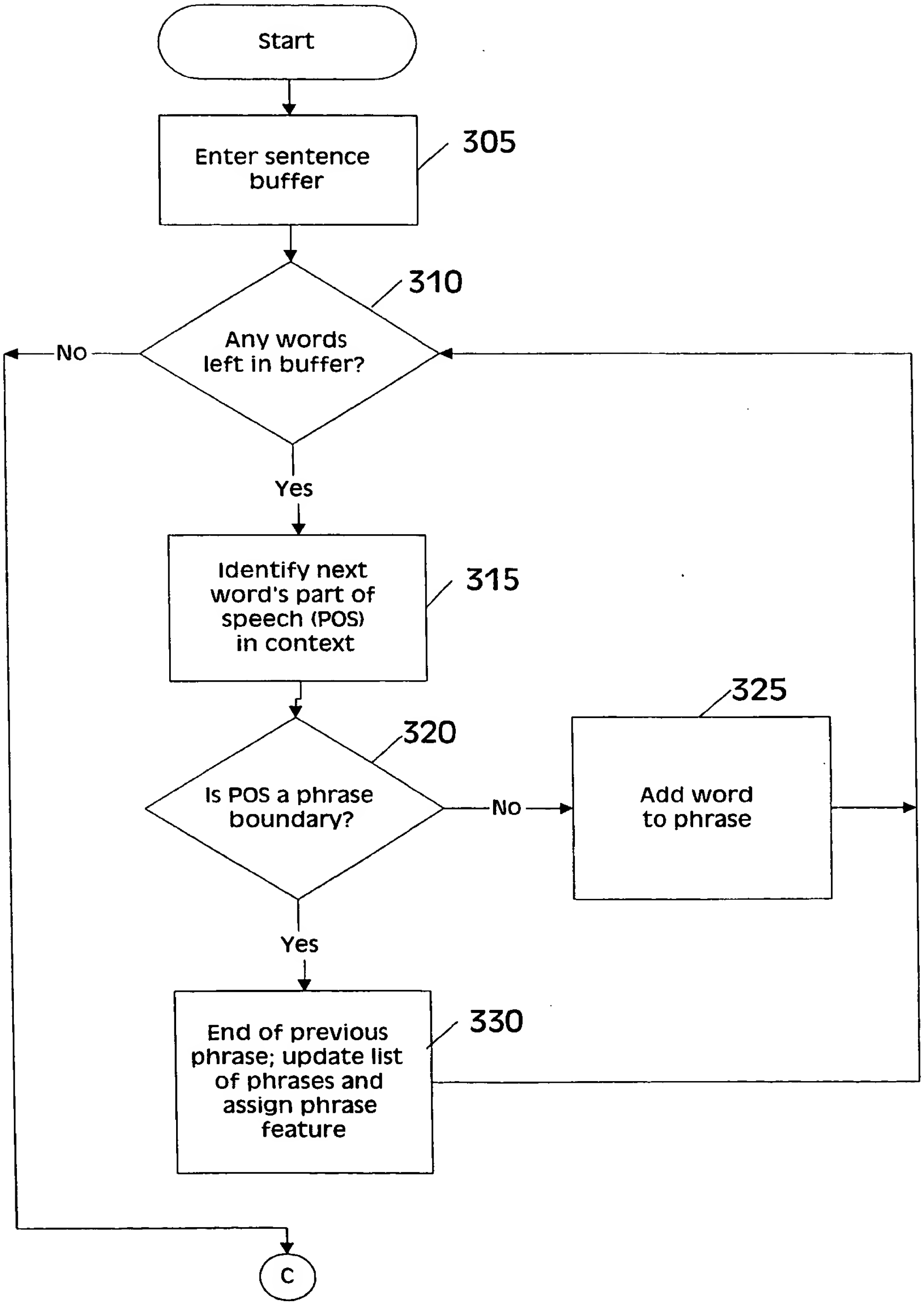

Start
Enter sentence buffer
305
Any words left in buffer?
310
No
Yes
Identify next word's part of speech (POS) in context
315
Is POS a phrase boundary?
320
No
Add word to phrase
325
Yes
End of previous phrase; update list of phrases and assign phrase feature
330
C

Fig. 3B

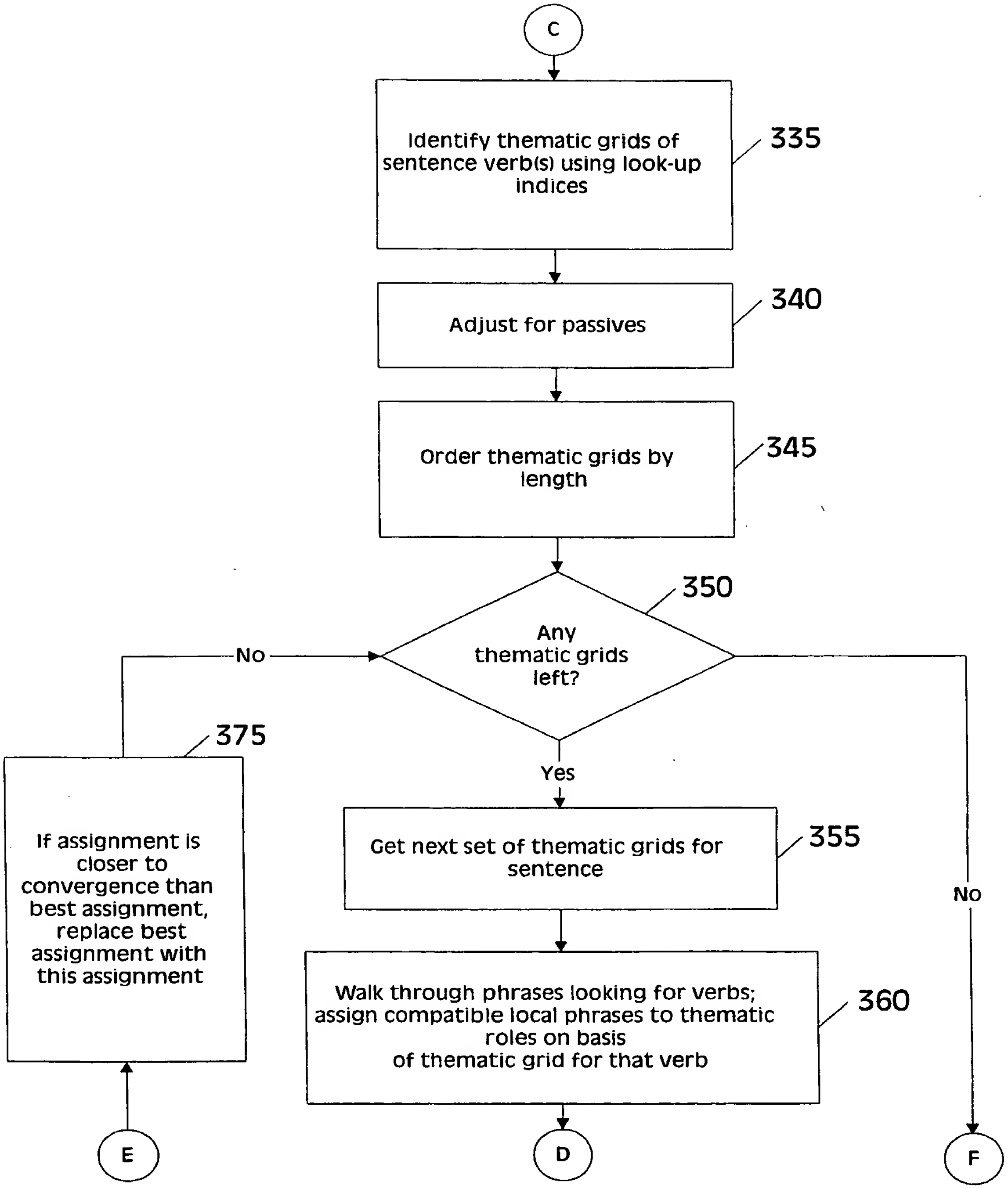

C
Identify thematic grids of sentence verb(s) using look-up indices
335
Adjust for passives
340
Order thematic grids by length
345
Any thematic grids left?
350
No
Yes
No
Get next set of thematic grids for sentence
355
Walk through phrases looking for verbs; assign compatible local phrases to thematic roles on basis of thematic grid for that verb
360
If assignment is closer to convergence than best assignment, replace best assignment with this assignment
375
E
D
F

Fig. 3C

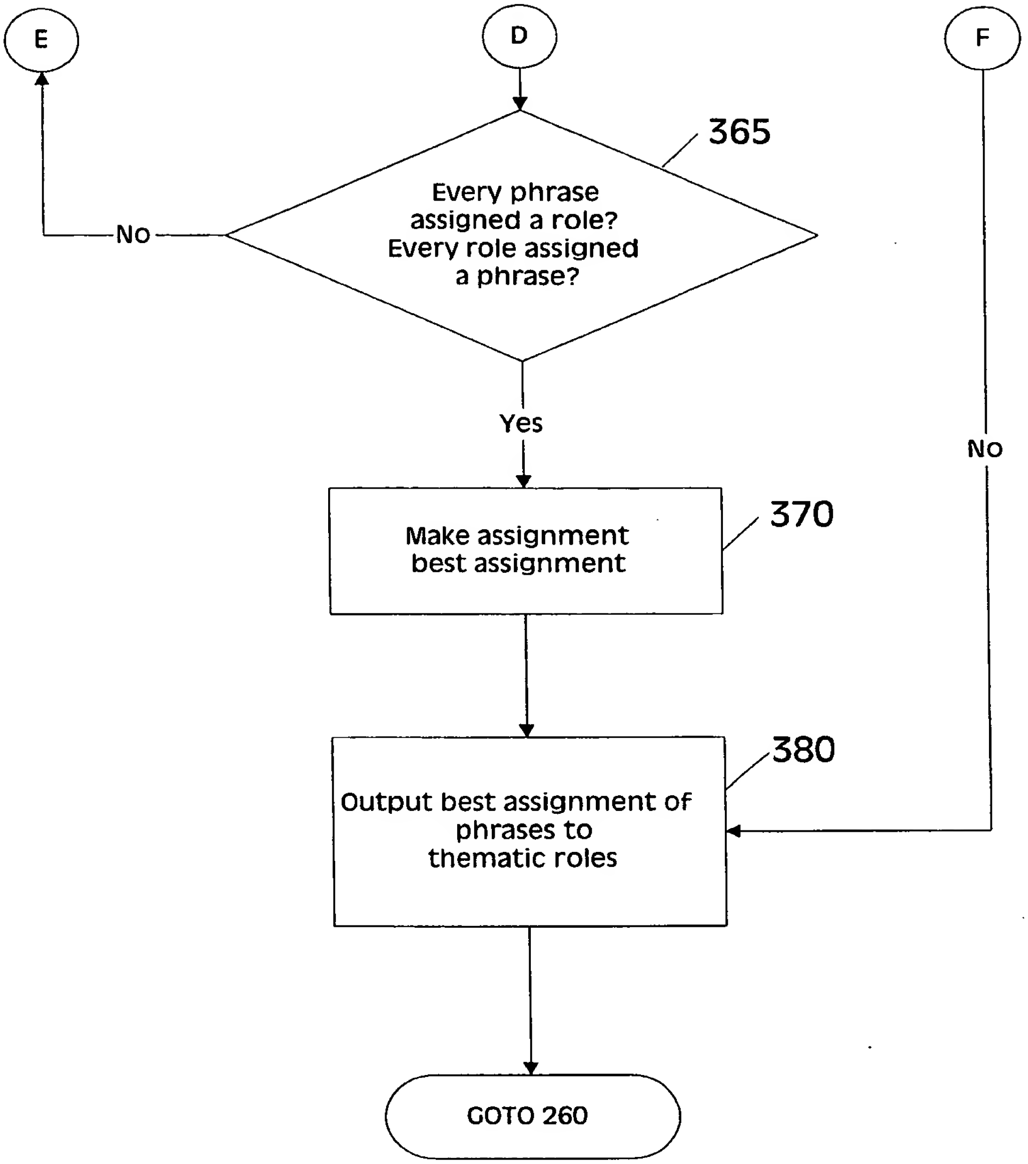
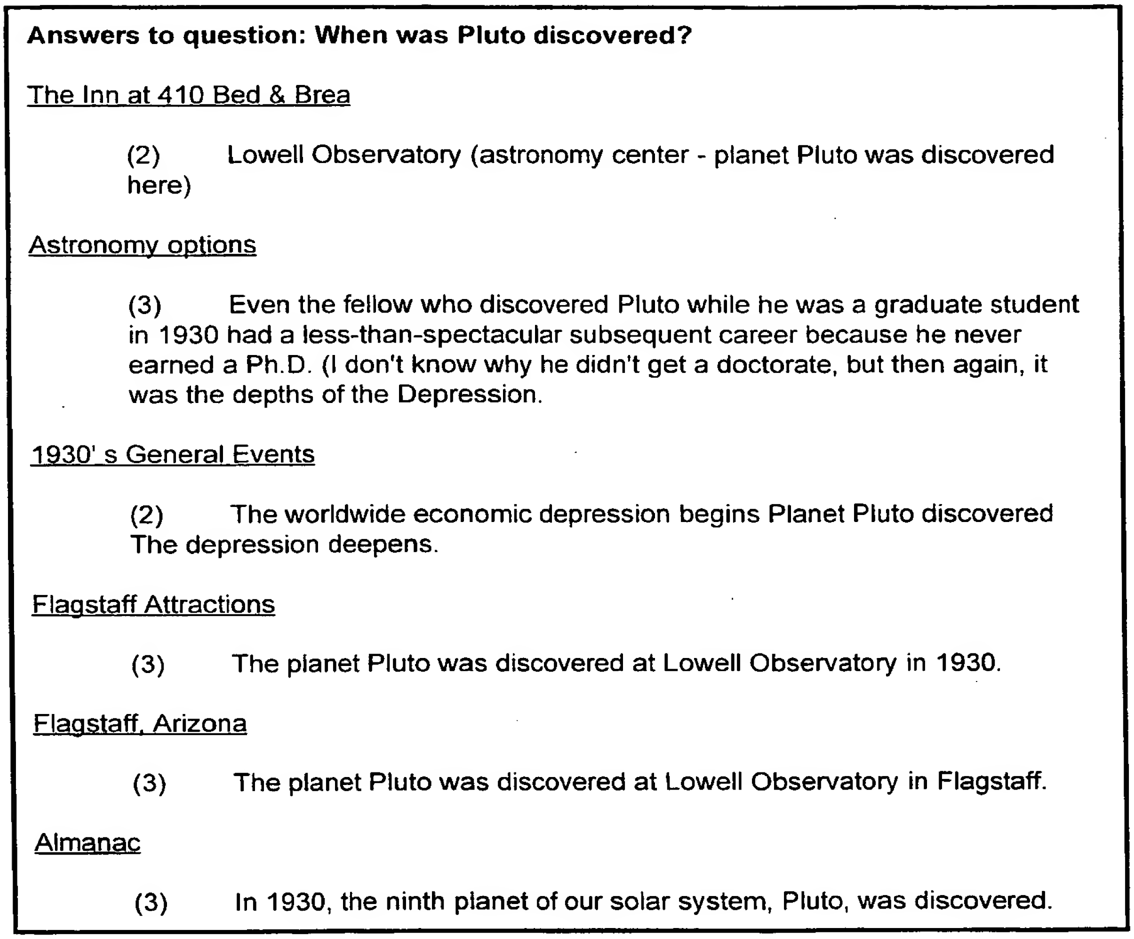

E
D
F
365
Every phrase assigned a role? Every role assigned a phrase?
No
Yes
No
370
Make assignment best assignment
380
Output best assignment of phrases to thematic roles
GOTO 260

Fig. 4A



Answers to question: When was Pluto discovered?

The Inn at 410 Bed & Brea

(2) Lowell Observatory (astronomy center - planet Pluto was discovered here)

Astronomy options

(3) Even the fellow who discovered Pluto while he was a graduate student in 1930 had a less-than-spectacular subsequent career because he never earned a Ph.D. (I don't know why he didn't get a doctorate, but then again, it was the depths of the Depression.

1930' s General Events

(2) The worldwide economic depression begins Planet Pluto discovered The depression deepens.

Flagstaff Attractions

(3) The planet Pluto was discovered at Lowell Observatory in 1930.

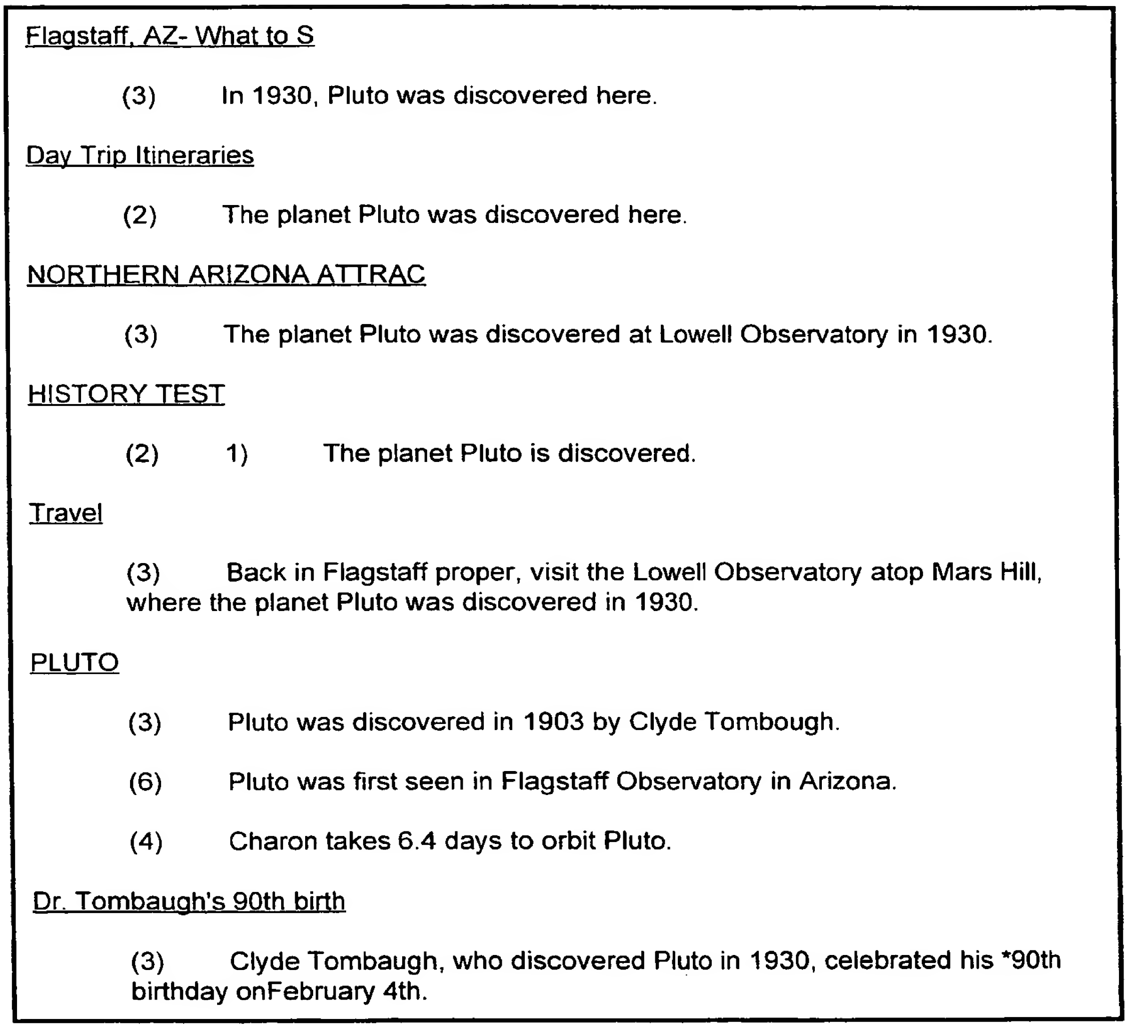
Flagstaff, Arizona

(3) The planet Pluto was discovered at Lowell Observatory in Flagstaff.

Almanac

(3) In 1930, the ninth planet of our solar system, Pluto, was discovered.

Fig. 4B



Flagstaff, AZ- What to S

(3) In 1930, Pluto was discovered here.

Day Trip Itineraries

(2) The planet Pluto was discovered here.

NORTHERN ARIZONA ATTRAC

(3) The planet Pluto was discovered at Lowell Observatory in 1930.

HISTORY TEST

(2) 1) The planet Pluto is discovered.

Travel

(3) Back in Flagstaff proper, visit the Lowell Observatory atop Mars Hill, where the planet Pluto was discovered in 1930.

PLUTO

(3) Pluto was discovered in 1903 by Clyde Tombough.

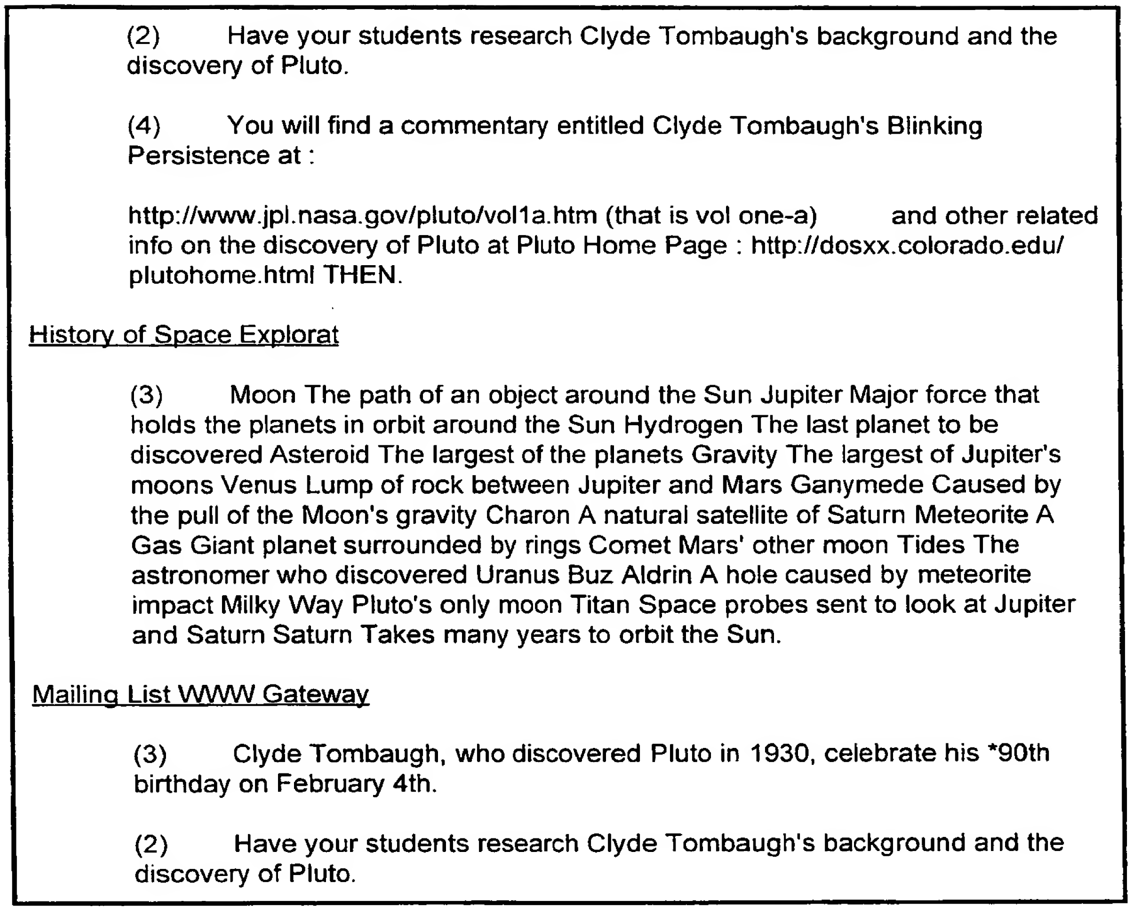
(6) Pluto was first seen in Flagstaff Observatory in Arizona.

(4) Charon takes 6.4 days to orbit Pluto.

Dr. Tombaugh's 90th birth

(3) Clyde Tombaugh, who discovered Pluto in 1930, celebrated his *90th birthday onFebruary 4th.

Fig. 4C



(2) Have your students research Clyde Tombaugh's background and the discovery of Pluto.

(4) You will find a commentary entitled Clyde Tombaugh's Blinking Persistence at :

http://www.jpl.nasa.gov/pluto/vol1a.htm (that is vol one-a) and other related info on the discovery of Pluto at Pluto Home Page : http://dosxx.colorado.edu/ plutohome.html THEN.

History of Space Explorat

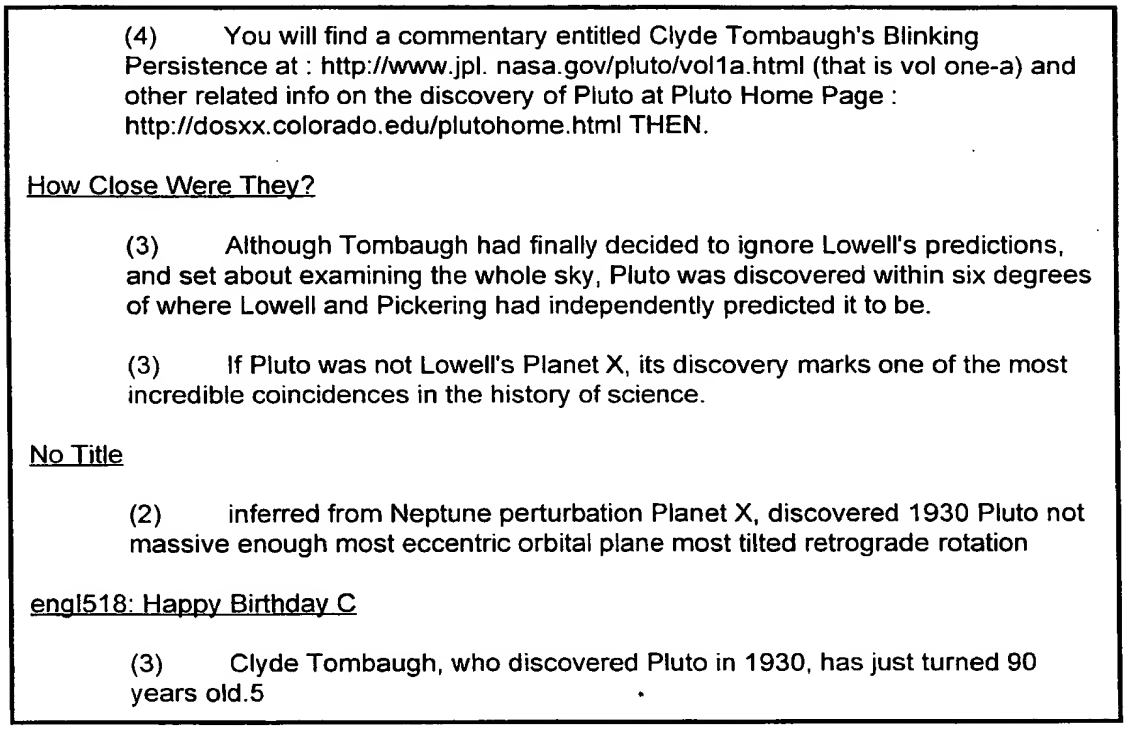
(3) Moon The path of an object around the Sun Jupiter Major force that holds the planets in orbit around the Sun Hydrogen The last planet to be discovered Asteroid The largest of the planets Gravity The largest of Jupiter's moons Venus Lump of rock between Jupiter and Mars Ganymede Caused by the pull of the Moon's gravity Charon A natural satellite of Saturn Meteorite A Gas Giant planet surrounded by rings Comet Mars' other moon Tides The astronomer who discovered Uranus Buz Aldrin A hole caused by meteorite impact Milky Way Pluto's only moon Titan Space probes sent to look at Jupiter and Saturn Saturn Takes many years to orbit the Sun.

Mailing List WWW Gateway

(3) Clyde Tombaugh, who discovered Pluto in 1930, celebrate his *90th birthday on February 4th.

(2) Have your students research Clyde Tombaugh's background and the discovery of Pluto.

Fig. 4D



(4) You will find a commentary entitled Clyde Tombaugh's Blinking Persistence at : http://www.jpl. nasa.gov/pluto/vol1a.html (that is vol one-a) and other related info on the discovery of Pluto at Pluto Home Page : http://dosxx.colorado.edu/plutohome.html THEN.

How Close Were They?

(3) Although Tombaugh had finally decided to ignore Lowell's predictions, and set about examining the whole sky, Pluto was discovered within six degrees of where Lowell and Pickering had independently predicted it to be.

(3) If Pluto was not Lowell's Planet X, its discovery marks one of the most incredible coincidences in the history of science.

No Title

(2) inferred from Neptune perturbation Planet X, discovered 1930 Pluto not massive enough most eccentric orbital plane most tilted retrograde rotation

engl518: Happy Birthday C

(3) Clyde Tombaugh, who discovered Pluto in 1930, has just turned 90 years old.5

INTERNATIONAL SEARCH REPORT

Interi. onal Application No
PCT/US 97/22943

A. CLASSIFICATION OF SUBJECT MATTER
IPC 6 G06F17/30

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 6 G06F

Documentation searched other than minimumdocumentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	EP 0 631 244 A (XEROX CORP) 28 December 1994	1-3,25, 26,28-30
A	see page 3, line 1 - page 11, line 45	5-9, 18-22, 31-33

A	RAYNER M ET AL: "Temporal relations and logic grammars" ECAI '86. 7TH EUROPEAN CONFERENCE ON ARTIFICIAL INTELLIGENCE. PROCEEDINGS, BRIGHTON, UK, 21-25 JULY 1986, 1986, LONDON, UK, CONFERENCE SERVICES, UK, pages 9-14 vol.2, XP002062196 see the whole document ---	1-3, 5-26, 28-33
A	EP 0 610 760 A (TOKYO SHIBAURA ELECTRIC CO) 17 August 1994 see column 13, line 22 - column 18, line 8 --- -/--	1,25

☒ Further documents are listed in the continuation of box C.

☒ Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publicationdate of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

1

Date of the actual completion of theinternational search 15 April 1998	Date of mailing of the international search report 04/05/1998
Name and mailing address of the ISA European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Authorized officer Fournier, C

INTERNATIONAL SEARCH REPORT

Interi nal Application No
PCT/US 97/22943

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category ˚	Citation of document, with indication,where appropriate, of the relevant passages	Relevant to claim No.
A	BALDAZO R: "NAVIGATING WITH A WEB COMPASS" BYTE, vol. 21, no. 3, 1 March 1996, page 97/98 XP000600179 see the whole document ---	1-4, 25-27
A	QUINTANA Y ET AL: "Graph-based retrieval of information in hypertext systems" SIGDOC '92. THE 10TH ANNUAL INTERNATIONAL CONFERENCE. CONFERENCE PROCEEDINGS. GOING ONLINE. THE NEW WORLD OF MULTIMEDIA DOCUMENTATION, PROCEEDINGS OF SIGDOC '92: 10TH ANNUAL ACM CONFERENCE ON SYSTEMS DOCUMENTATION, OTTAWA, ONT., CANADA, 13-16 OCT. 19, ISBN 0-89791-532-1, 1992, NEW YORK, NY, USA, ACM, USA, pages 157-168, XP002062197 see the whole document -----	1,25

1

INTERNATIONAL SEARCH REPORT

ıııformation on patent family members

Interı. ınal Application No
PCT/US 97/22943

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP 0631244 A	28-12-94	US 5519608 A JP 7056954 A	21-05-96 03-03-95
EP 0610760 A	17-08-94	JP 6231178 A JP 7182373 A	19-08-94 21-07-95